

PRICE  
.75

\*\*\*\*\*  
\*  
\* SMUG BYTES \*  
\* Volume 4, Number 4 \*  
\* April 1987 \*  
\*  
\* Subscriptions = \$10.00/year \*  
\* or free with swap. Send to: \*  
\*  
\* SINCLAIR MILWAUKEE USERS GROUP \*  
\* P.O. Box 101 Butler, WI 53007 \*  
\*  
\* ----- \*  
\*  
\* THIS MONTH: \*  
\*  
\* -March Meeting Notes \*  
\*  
\* -256K AERCO Interface \*  
\*  
\* -Presidents Message \*  
\*  
\* -  
\*  
\* -And Other Great Things \*  
\*  
\*  
\*-----  
\* NEXT MEETING DATE: 5/13/87  
\*-----  
\*  
\* Send all contributions by the \*  
\* first day of the month to:  
\*  
\* Bill Heberlein  
\* Editor  
\* SMUG BYTES  
\* 5052 N. 91st Street.  
\* Milwaukee, WI 53225  
\*  
\*\*\*\*\*

## 1987 OFFICERS & WORKING MEMBERS

President - Neal Schultz  
- 353 4522  
V. Pres. - W. Heberlein  
- 527 2191  
Secretary - Lloyd Dreger  
- 321 0694  
Treasurer - J. Frohboese  
- 377 1385  
Education - Dick Cultice  
- 542 3591  
  
- Librarians  
Tape - Ed Pawloski  
- 1 338 0260  
Book - Dennis Nickels  
- 1 284 9691  
  
- Sub Group Leaders  
  
Games - Neal Schultz  
- 353 4522  
Meeting date 3rd Sat. of the  
month.  
  
Hardware - Gordy Kraemer  
- 421 0179  
No set date. Call for info.  
  
Spectrum - Rudy Hilsmann  
- 251 5291  
Meeting date 3rd Wed. of the  
month.  
  
RP/M - Dick Cultice  
- 542 3591  
Meeting date 4th Wed. of the  
month.

## AT OUR LAST MEETING--

There was some good discussion on our classes and what classes will be held. There will not be a basic Basic class unless more of you are interested. There will be Advanced Basic and Intro to Machine Code. Special class will be the Modem Class. The first two classes will be taught by Lloyd Dreger and the Modem class will be taught by Dave Franson. If interested See Dick Cultice for signup sheets.

There was more discussion dealing with the COMPUTERFEST in May and what was happening there. SMUG will be giving two seminars. The first will be Dave Franson with a Graphics presentation. The second will be Dick Cultice, Dave, again, and myself. This presentation will be on printers and their use.

President, Neal Schultz asked if both seminars could be presented at the next meeting. Because of the length, 50 min., it was decided to present shortened versions. This should be a very informative meeting. If you can't make it watch for future serials of both in this newsletter.

The final bit of business was the creation of a digitizer for the TS068. Gordon Kraemer, Hardware Hacker, will look into what is required to build one. What this means is that the TS2068 can take a picture from a TV screen and copy it to the computer. The only requirement would be that the picture must be still while the computer "reads" the screen. So I think it will require a video recorder. IF THIS WORKS just think of the pictures that can be stored in the memory of the computer. You can store family, homes or anything you have on your VCR.

## Upcomming events:

Saturday and Sunday May 2 and 3, 1987 is the COMPUTERFEST in Indiana. See last months SMUG Bytes.

## NEW CONTEST:

In somewhere in SMUG Bytes will be a page with different types of code. The solution to the message must be postmarked or in my hands by the 5th of May. If there is more than one correct respondent I will have a drawing at the May meeting. Decision by the judges, me, will be final.

The prize will be your choice of 2 rolls of 2040 paper, 3 5 1/4 inch DSDD disc's or 100 pages of 9 1/2 x 11 printer paper. The answer will appear in the June Issue of SMUG Bytes. I want each line printed out under the code. Hint the puzzle should have been printed in the February issue.



## ACTIVATING ARECO BANKS FOR BASIC

BY LLOYD DREGER

Here is the program those of us with Aerco Disk drives with extra banks of memory have been waiting for. Although we had the code for putting basic into the dock bank since June (See FD-68 vol 1.2) albeit with an error, we were missing the necessary bank numbers for the other banks. Many thanks to David Hill for the 4 drive program modifications. I just put it all together.

The two port numbers to remember are 60639 for a 2 drive system and 62687 for the 4 drive system. The bank numbers for the 2 drive system are 128, 132, 136, and 140 AND 128, 129, 136 and 137 for the 4 drive system. You just have to remember the numbers that apply to your particular system.

To install the correct codes into the other banks you will have to type in two short m/c programs and a basic program. What is listed below is for the 2 DRIVE SYSTEM.

### Program 1:

```
1 REM 2 DRIVES
5 FOR A = 24047 TO 24133
10 READ B: POKE A,B: NEXT A
20 DATA
243,62,3,211,244,33,0,0,17,0,128,1,
0,64,237,176,62,3,
211,244,62,132,1,223,236,237,121,33
,0,128,17,0,0,1,0,237,176,
62,136,1,223,236,237,121,33,0,128,1
7,0,0,1,0,64,237,176,62,140,
1,223,236,237,121,33,0,128,17,0,0,1
,0,64,237,176,62,128,1,223,
236,237,121,62,1,211,244,251,201
```

Check the data line carefully and run the program then save this code to disk with MOVE "banks.bin",24047,87.

## ANNOUNCING THE GRAND OPENING OF

# CSI

Milwaukee's New Full-Service Computer Center

- Sale and Lease of New and Used Computers
- Used IBM Systems and Peripherals
- New IBM-Compatible Systems and Peripherals
- Extensive Selection of Software and Supplies
- Complete Service and Repair
- Low Prices
- Friendly People

**COMPUTER SYSTEMS, INC.**

11710 West North Avenue  
Wauwatosa, Wisconsin 53226

**771-2983**



Your Computer Experts

### Program 2:

```
1 REM BASIC INSTALLER:  
5 FOR A = 24320 TO 24516  
10 READ B: POKE A,B: NEXT A  
20 DATA  
243,219,244,8,58,158,95,6,0,184,40,  
62,8,6,1,184,40,28,  
33,0,92,17,0,128,1,203,0,237,176,62  
,1,211,244,33,0,91,17,0,92,1  
30 DATA  
203,0,237,176,251,201,33,0,92,17,0,  
91,1,203,0,237,176,  
62,241,211,244,33,0,128,17,0,92,1,2  
03,0,237,176,251,201,62,1,211  
40 DATA  
244,50,158,95,33,0,92,17,0,91,1,203  
,0,237,176,62,241,  
211,244,33,204,128,54,128,35,54,13,  
35,54,128,33,89,92,17,75,92,1  
50 DATA  
83,92,62,204,54,205,18,2,35,19,3,62  
,128,119,18,2,24,  
142,33,0,91,17,0,128,1,203,0,237,17  
6,205,2,10,33,0,128,17,0,91,1  
60 DATA  
203,0,237,176,201,0,1,241,17,0,0,33  
,0,0,1,0,0,58,159,  
95,211,244,126,8,58,160,95,211,244,  
8,18,19,35,11,120,177,32,235  
70 DATA 58,160,95,211,244,201
```

Again, check the data lines carefully, run the program and save the code to disk with MOVE "dockbas.bin",24320,197.

### Program 3:

```
1 REM 2 DRIVES  
2 REM FOR 4 DRIVE SYSTEMS CHANGE  
POKE 60639 TO POKE 62687 USING 128,  
129, 136 AND 137 FOR BANK NUMBERS.  
ALSO POKE 24071, 24089, 24107 AND  
24125 WITH 244. THEN POKE  
24068,129: POKE 24086,136: POKE  
24104,137  
5 CAT "BANKS.BIN",  
10 BORDER 1: PAPER 1: INK 5: CLS  
15 OUT 244,1: OUT 60639,128: PRINT  
"NOW TRANSFERRING OPERATING SYSTEM  
TO SHADOW DOCK BANKS""  
20 RANDOMIZE USR 24320
```

```
25 FOR B = 128 TO 140 STEP 4:REM  
CHANGE FOR 4 DRIVE SYSTEM  
30 OUT 60639,B:PRINT "OUT 60639,  
";B:OUT 244,1:POKE 37,B: NEXT B:REM  
INSTALL BANK NUMBERS  
35 OUT 244,1  
40 PRINT "ALL BANKS NOW HAVE AERCO  
DISK DRIVE INTERFACE OPERATING  
SYSTEM WITH ID AT BYTE 37"  
42 PRINT  
45 CAT "DOCKBAS.BIN".  
50 INPUT "WHICH BANKS DO YOU WISH  
BASIC SYSTEM INSTALLED? (0-3)  
0=DOCK ";A  
55 IF A <0 OR A >3 THEN GO TO 50  
60 LET A = A*4: REM FOR 4 DRIVE  
SYSTEM USE--IF A >2 THEN LET A = A  
+ 6  
65 LET A = A + 128  
70 OUT 60639,A: RANDOMIZE USR  
24320: OUT 244,1  
75 PRINT "BASIC SYSTEM IN BANK ";A  
80 INPUT "INSTALL BASIC SYSTEM IN  
ANOTHER BANK? (Y/N) ";Y$  
85 IF Y$ = "Y" OR Y$ = "y" THEN  
POKE 24478,0: GO TO 50  
90 CLS: PRINT TAB 8: FLASH 1: "READ  
CAREFULLY": FLASH 0  
100 PRINT ""YOU ARE STILL IN THE  
HOME BANK BUT ARE USING CHUNK 0 OF  
THE LAST BANK YOU ACTIVATED WITH  
BASIC."  
105 PRINT "TO ACTIVATE THIS BANK  
MERELY DO:"; FLASH 1; "RANDOMIZE USR  
24320: OUT 244,241"; FLASH 0  
110 PRINT "TO ACTIVATE BASIC IN  
DIFFERENT BANK DO: "; FLASH 1; "OUT  
24320,#: "; FLASH 0; "WHERE # IS  
ONE OF THE BANK NUMBERS GIVEN. THEN  
FOLLOW WITH: RANDOMIZE USR 24320:  
OUT 244,241 ALL ON THE SAME LINE."  
120 PRINT "TO GET BACK TO HOME BANK  
DO: RANDOMIZE USR 24320 AND OUT  
244,1"  
130 PRINT "TAB 4: FLASH 1: "PRESS  
ANY KEY FOR MORE"; FLASH 0:PAUSE 0  
140 CLS: PRINT "ANY BASIC PROGRAM  
CAN BE LOADED DIRECTLY INTO AN  
ACTIVATED BANK BUT ""COPY""  
COMMANDS MUST BE CHANGED TO  
RANDOMIZE USR 24452"  
150 PRINT "BINARY CODE PORTIONS OF  
PROGRAMS MUST BE FIRST LOADED INTO  
THE HOME BANK AND TRANSFERRED TO
```

THE DESIRED BANK WITH THE CODE AT 24479 AS LISTED IN THIS PROGRAM. ONCE LOADED, A SAVE TO THE DISK THEN ALLOWS DIRECT LOADING FROM BASIC USING THE NEW BIN SAVE."

```

160 PRINT '"NEVER USE THE ""NEW"" COMMAND TO ERASE A PROGRAM AS THE COMPLETE INSTALLATION ALSO IS DELETED. USE DELETE #,# INSTEAD."
170 PRINT 'TAB 4; FLASH 1; "PRESS ANY KEY FOR MORE"; FLASH 0: PAUSE 0
180 CLS: PRINT "THIS PROGRAM USES CODE AT 24047 TO 24135 AND 24320 TO 24516. THE BASIC PROGRAM MAY NOW BE ERASED. YOU MAY WISH TO USE THE LINES AT 9000 FOR YOUR FORGRAM SETUPS AFTER WHICH THEY TOO CAN GO."
190 PRINT """TAB 7;"HAPPY COMPUTERING"
200 STOP
9000 REM XFER BIN
9005 INPUT "WHICH BANK? 128, 132, 136, 140? ";A: REM 128, 129, 136, 137 FOR 4 DRIVES
9010 POKE 60639,A: REM 62687 FOR 4 DRIVE
9015 INPUT "ENTER FILE NAME TO BE TRANSFERRED ";A$: LET B$ = A$
9020 LET A = (A - 128)/4 : REM "A = A - 128 : IF A > 2 THEN LET A = A - 4" FOR 4 DRIVE SYSTEM
9025 IF LEN B$ > (10 - A) THEN LET B$ = B$( TO 10 - A)
9030 FOR X = 1 TO A: LET B$ = B$ + "*": NEXT X: REM NAME IS SUFFIXED WITH 1 TO 4 * TO SPECIFY WHICH BANK
9035 LET A$ = A$ + ".BIN"";: CAT "A$",
9040 INPUT "STARTING ADDRESS ";A: LET S = A: GO SUB 9100
9045 INPUT "LENGTH OF CODE (MAX 32768) ";B: IF B > 32768 THEN PRINT "CODE TOO LONG FOR BANK": STOP
9050 LET S = B: GO SUB 9110
9055 RANDOMIZE USR 24481
9060 LET B$ = B$ + ".BIN"";+ STR$ A + "," + STR$ B: MOVE "B$",
9065 OUT 244,1: STOP
9100 GO SUB 9200: POKE 24485,LO: POKE 24486,HI: POKE 24482,LO: POKE 24483,HI: RETURN

```

```

9110 GO SUB 9200: POKE 24488,LO: POKE 24489,HI: RETURN
9200 LET HI = INT (S/256): LET LO = S - (HI*256): RETURN

```

Save this program to disk with MOVE "banks.bas", before running it.

Before running the program, let's explain what is going to happen. The first code actually takes chunks 0 and 1 and loads them in the home bank above 32768. It then activates all 4 banks in turn and writes the code to these banks in chunks 0 and 1 returning to the home bank. In the process it also writes the bank # into byte 37 of each bank. A peek of this byte will always tell you what bank is ready.

The code of the 2nd program sets up the basic programmer into each bank you select. Certain things are necessary to run basic in a bank. Since each program will be a different length a copy of the systems variables table must be preserved somewhere before leaving the home bank. This is done by storing it into the printer buffer for the 2040 printer. A new copy of the system variables is then set up with a Program starting address of 32972 with the proper end markers inserted into the ram for the end of basic, the end of vars and eline. The space between 32768 and 32972 stores the 204 bytes of the system variables table for that bank. Upon shifting banks the proper system variables are put into the normal systems variable area of chunk 2 which also contains the display file for the monitor/tv. Doing all the OUT 244,241's activates all but chunks 1, 2 and 3 of each bank. Never activate chunk 2 or you lose the display file and the systems variable table. Never activate chunk 3 or you lose the machine stack and the exrom switching code.

Because the printer buffer is used, a special routine for the extra banks is used to copy out the printer buffer to another area so it can be used for the 2040 printer from the extra banks--that is why the Randomize Usr 24452.

Also note that the machine code stack is common to all banks. This gives us a headache in basic. We can shift banks in the middle of a program as long as we are not in a subroutine when we do it--a subroutine pushes the return address unto the stack and it may get buried or be used as the wrong time if we are not careful.

The expert will note that only the basic 32 character per line screen can be used at this point. Further modifications are necessary before you activate other screen modes.

Also note at this time that there is no transferring of basic variables from one program to another. Each bank has its own basic variables table.

Now for a little demo. Run the program and install basic in as many banks as you desire. Read all the help screens. When the program stops enter in command mode the RANDOMISE USR 24320 and Out 244,241. Do a CLS and try LIST---you get a 0 0 ok in the bottom corner but no program: Now, just for fun type in the line 1 PRINT PEEK 37 and RUN it just to make sure the bank basic works. Okay? Let's go back to the home bank. RANDOMIZE USR 24320 and OUT 244,1. Doing a LIST should get back a listing of this program. For a Different bank use OUT 60639,# where # is a different bank you have set up for basic. Once again RANDOMISE USR 24320 and OUT 244,241 followed by CLS and LIST and once again no program. Enter something different for a one line program if

you like and do back to the home bank again. Note that each use of RANDOMISE USR 24320 toggles you between the home bank and the activated bank. Don't forget the OUT 244 statements or your computer will get lost. Never use NEW or all this initialization you have been doing is lost.

What you do with all 5 banks in basic mode only you and your wildest imagination can decide. Have fun.

The actual routines are:

#### Operating System:

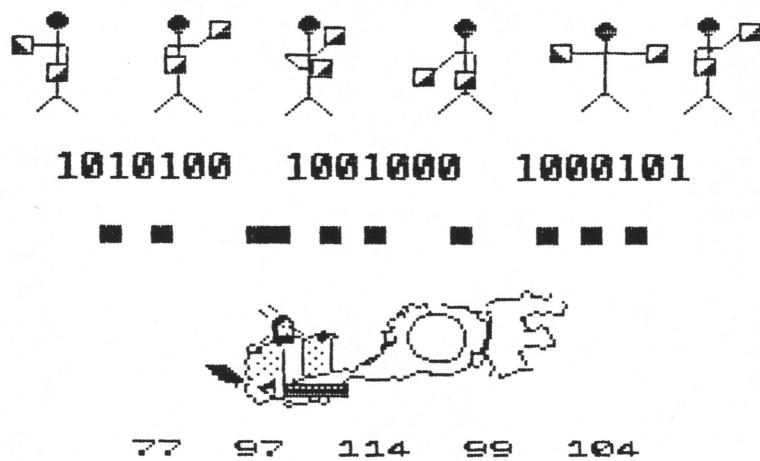
24047 F3	DI
24048 3E03	LD A, 3
24050 D3F4	OUT (244),A
24052 210000	LD HL, 0
24055 110000	LD DE, 32768
24058 010040	LD BC, 16384
24061 EDB0	LDIR
24063 3E03	LD A, 3
24065 D3F4	OUT (244), A
24067 3E84	LD A, 132
24069 01DFEC	LD BC, 60639
24072 ED79	OUT (C), A
24074 210080	LD HL, 32768
24077 110000	LD DE, 0
24080 010040	LD BC, 16384
24083 EDB0	LDIR
24085 3E88	LD A, 136
24087 01DFEC	LD BC, 60639
24090 ED79	OUT (244), A
24092 210080	LD HL, 32768
24095 110000	LD DE, 0
24098 010040	LD BC, 16384
24101 EDB0	LDIR
24103 3E8C	LD A, 140
24105 01DFEC	LD BC, 60639
24108 ED79	OUT (244), A
24110 210080	LD HL, 32768
24113 110000	LD DE, 0
24116 010040	LD BC, 16384
24119 EDB0	LDIR
24121 3E80	LD A, 128
24123 01DFEC	LD BC, 60639
24126 ED79	OUT (C), A
24128 3E01	LD A, 1
24130 D3F4	OUT (244), A

24132 FB	EI	24398 329E5F	LD (24478), A
24133 C9	RET	24401 21005C	LD HL, 23552:
SAVE HOME SYS VARS			
24404 11005B LD DE, 23396			
24407 01CB00 LD BC, 203			
24410 EDBO LDIR			
24412 3EF1 LD A, 241:			
CHANGE TO BANK			
24414 D3F4 OUT (244), A			
24416 21CC80 LD HL, 32972:			
ADJUST PROG. VARS & ELINE			
24419 3680 LD (HL), 128			
:END MARKERS			
24421 23 INC HL			
24422 3680 LD (HL), 128			
24424 23 INC HL			
24425 3680 LD (HL), 128			
24427 21595C LD HL, 23641			
24430 114B5C LD DE, 23627			
24433 01535C LD BC, 23635			
24436 3ECC LD A, 204			
24438 36CD LD (HL), 205			
24440 12 LD (DE), A			
24441 02 LD (BC), A			
24442 23 INC HL			
24443 13 INC DE			
24444 03 INC BC			
24445 3E80 LD A, 128			
24447 77 LD (HL), A			
24448 12 LD (DE), A			
24449 02 LD (BC), A			
24450 18BE JR HOME			
24452 21005B COPY LD HL, PRINT			
BUFFER			
24455 110080 LD DE, BANK			
STORAGE			
24458 01CB00 LD BC, 203			
24461 EDBO LDIR			
24463 CD020A CALL 2562			
:COPY			
24466 210080 LD HL, BANK			
STORAGE			
24469 11005B LD DE, PRINT			
BUFFER			
24472 01CB00 LD BC, 203			
24475 EDBO LDIR			
24477 C9 RET			
24478 00 NOP : INIT			
FLAG			
24479 01 XFER CODE SOURCE BANK			
CHUNKS TO BE ENABLED			
24480 F1 DESTINATION			
BANK CHUNKS TO BE ENABLED			
24481 110000 LD DE,			
INIT LD A, 1: SWICH			
TO HOME AND SET INIT BYTE			
24396 D3F4 OUT (244), A			

DESTINATION ADDRESS--SEE NOTES  
 24484 210000 LD HL, SOURCE  
 ADDRESS  
 24487 010000 LD BC, LENGTH  
 24490 3A9F5F LOOP LD A, (24479):  
 LD SOURCE BANKS CHUNKS  
 24493 D3F4 OUT (244), A:  
 ENABLE SOURCE BANKS  
 24495 7E LD A, (HL)  
 :LOAD BYTE FOR XFER  
 24496 08 EX AF,AF':STORE  
 IT  
 24497 3AA05F LD A, (24480):  
 GET DEST CHUNKS  
 24500 D3F4 OUT (244),  
 A:ENABLE DESTINATION  
 24502 08 EX AF,AF' :GET  
 BYTE BACK  
 24503 12 LD (DE), A :PUT  
 IN DEST  
 24504 13 INC DE  
 :INCREMENT REGISTERS TO NEXT BYTE  
 24505 23 INC HL  
 24506 0B DEC BC :REDUCD  
 CNTER AND CHECK FOR ZERO  
 24507 78 LD A, B  
 24508 B1 OR C  
 24509 20EB JR NZ, LOOP  
 24511 3AA05C LD A,  
 (24480):ENABLE DESTINATION BANK  
 24514 D3F4 OUT (244), A  
 24516 C9 RET

NOTES: This program will only work between the home bank and another bank not two different extra banks. DE must be the destination address in the extra bank and HL the source address in the home bank--generally they are the same address. BC is the number of bytes to transfer. Use normal lo/hi notation.

The basic of program 3, lines 9000-9200 asks for the bin name without suffix, loads it into the home bank, transfers it to the enabled extra bank and then saves a copy from the extra bank to disk with 1 to 4 astericks in the name to indicate which bank it was saved from. In other words, it's very user friendly.



## FOR SALE:

If you have anything for sale, you can advertise it for free in this newsletter (provided you're a SMUG member). Our newsletter reaches an ever-growing number of TS User Groups throughout the country, increasing your chances of a sale.

## UNLEASH THE POWER OF EXTENDED COLOR MODE ON YOUR T/S 2068!!

EXTEND PAINT gives you eight times the normal color resolution.

Over 25 unique functions. Joystick control, menu-driven.

Extensive printer support, including hires, greyscale screen dumps to T/S 2040 and Epson compatibles, as well as FULL COLOR hires

dumps to Canon color ink-jet and compatibles.

Supports AERCO Centronics interface.

Cassette, with 24 page manual. \$19.95 pp.  
from:

Dave Franson  
3534A E.

Squire Ave.  
53110  
Cudahy, WI

## FOR SALE:

-continued-

Catalog your Aerco disks! Program compiles an alphabetic list of programs residing on all your Aerco disks. Prints the program name, data type (BAS, SCR, etc.), and the system name given the disk. Output to either the 2040 or full-size printer through Aerco CPI (modifiable for other printer interfaces). \$5.00 plus \$2.00 shipping and handling. Contact:

Bill Heberlein  
5052 N. 91st St.  
Milwaukee, WI 53225

Also available from Bill at the above address is a BOOT program for the Aerco disk drives. This program will bring up any disk it resides on upon powerup, and allows you to select a program by typing a single character. It can be had for \$2.00.

## MACHINE CODE BOOKS by Dr. Lloyd Dreger.

The T/S 2068 ROM Manuscript-- complete disassembly of the ROM with notes, comments, and cross-references to the Spectrum.

Introduction to 2068 Machine Code-- a self teaching manual for beginning machine code students, showing how BASIC is converted into code.

Either book \$16.95 (SMUG members \$15.00) or both for \$30.00 plus \$2.50 S&H. Wisconsin residents add 5% sales tax.

Available through:

SMUG,  
P.O. Box 101,  
Butler, WI 53007.

## " THE REAL THING "

A lot of pleasure and satisfaction can be had in doing and finding ways to use your computer. At our last CPM meeting we tried to get a 232 port up and operating in CPM. We were side tracked when Modem753 was loaded into one computer and Mterm was loaded into a second computer. What followed was a down load from CPM to the Sinclair system using Xmodem. Here is how it was done.

The telephone cables were connected using a double phone receptical with 12VDC across the line. A 200 ohm resistor was used to limit the current. After loading Mterm with Xmodem in one of the computers and Modem753 into the other, we began.

The first thing we had to do was set the protocol. This is done from the menu in Mterm and by answering the prompts in Modem753. This was the protocol used.

	MTERM II	MODEM753
DUPLEX:	FULL	* E
PARITY:	NONE	NONE
WORD SIZE:	8	8
STOP BITS:	1	1
CON:	NONE	-----

\* E was used to make the connection from CPM and Cap Shift-Symb1 Shift,M from Mterm in terminal mode.

After the exchange of a few clever phrases and the sending of some MACRO's from Mterm, a few more words from the CPM terminal to check everything out, we were ready for our first attempt at transmitting a file.

The fist try to transmit a file with Mterm in terminal mode, was not successful. CPM was looking for a check sum which is not provided by Mterm. The second try, Xmodem was used. This caused some excitement, for the file was being sent to the buffer in Mterm. The buffer now held a file which could be saved and used in just about any text editer.

Here is the procedure used:

With both terminals connected and the red lights on, each went to their respective MENU. The terminal using Mterm went to basic with the E command and type in the following. RANDOMIZE USR 54415 and waited the ok to continue. Meanwhile at the CPM consol the command S:file.DOC was typed in with a CR. The message came up WAITING and the ok was given to procede. The CR was pressed and the file was transmitted 128 bytes at a time. After each block was sent, a plus appeared from XMODEM and a hex number, (the number of the block being sent) from Modem753. We had succeeded.

Now you ask "What can this do for me?". CPM has a number of utilities that can be useful in writing and patching machine code programs with good documentation. You can write a new system or refine an old one to better suit your needs. Let your imagination come up with something new and useful and you will have made your mark in the computer field. We have the tools and they can be fun to use.

Next month we will explore the 232 port if the programming is available from AERCO. With the expanded CPM system, bank switching,new software and inventive hardware, the TIMEX 2068 will be not be forgotten as long as you and I support it.

If you are having a problem or would like a review of a subject about CPM write the news letter and let us know and we will try to cover your subject.

Richard

## WAR GAMES

FROM THE PRES.

24 DAYS AND COUNTING!

An anxious feeling is building up for those members planning on attending the 2nd annual computerfest. There is so much to do and so little time left to do it in.

It is surprising the enthusiasm still left for our little computer even after Timex left the market. But I guess the dedication of the users overcomes everything.

War simulation games are games of strategy and planning requiring no dexterity skills with a troublesome joystick and no fast reflex action.

War in the East, by Mark L. Streuber.  
WWII--Eastern Front, by Lloyd Dreger.

Both these games are a takeoff of the Avalon Hill game called Barbarosa after the German nickname of their opening Russian campaign. You play the Nazi German and Romanian (Axis) armies invading Russia in the Summer of 1941 with the object of capturing Moscow and holding it for 4 turns while the computer plays Russia. The entire game takes 2 to 4 hours for the 2068 game and 3 or 15 hours on the TS1000.

## GK ENTERPRISES

Your Computer Supply Source

No need to wait for a sale.

Our prices are low everyday

Wide selection of printer ribbons

AXIOM

C. ITOH

EPSON

OKIDATA

PANASONIC

STAR

AND MANY MORE.

\*\*\*\*\*  
\*SMUG SPECIAL\*  
\*\*\*\*\*

2700 sheets 9 1/2 x 11 20# Microperf \$23.50

\*\*\*\*\*  
\* 546-0111 \*  
\*\*\*\*\*

8634 W. National Avenue - West Allis WI 53227

Several years ago Mark sent SMUG a TS1000 copy of War in the East for review. It required 32k of memory and several major bugs were discovered. There are 4 playing maps. The movement of the Russian armies was straight across the screen and as armies reached the Baltic Sea they were killed off... sort of a "Lemmings to the sea" approach which left Moscow undefended which was not historically accurate. It took the computer 20 minutes to calculate the battle results, move the Russian armies, calculate their battle results and set up for the next German army move for the first turn and about 10 minutes for every turn after that.

Lloyd, a SMUG member, who had played the Avalon Hill game, decided to take up the challenge to correct the deficiencies and ended up rewriting the whole game adding a m/c routine to reduce map

printing time to less than 10 seconds. He even added a 5th map and corrected the intermap detection problem as well as the "lemming problem" while leaving Moscow always heavily guarded. The first turn wait was reduced to 3 to 4 minutes while succeeding turns were reduced to 1 minute or less. This made the game much harder to win, so he added two other variations to make the chances of winning higher for the amateur gamesman. And, he reduced the memory requirement to the 16k memory most TS1000 users have.

If you are interested in methods of reducing memory requirements, this is a real shoehorn job.

Fall of the Third Reich. Mark Steuber.

Mark gets better and better in his later attempts. You again play Germany in a hopeless situation of 1944 and the best you can do is delay the inevitable for a little while.

Ardennes. Mark Steuber.

The Germans in the Battle of the Bulge. Break through the allied lines, take as many towns as possible and then defend for all you are worth. Hard to win.

Britian Invaded. Mark Steuber.

What it would have been like if Germany had invaded England in WWII. Great graphics with a scrolling map and the ability to say which way you are attacking. You play England and will lose this game if you don't go on an aggressive attack. The German army movement leaves something to be desired. You must keep London at all cost and not let the Germans cut the rail lines. Arnhem. Robert T. Smith (Spectrum only at the present time)

The attempt by General Mongomery to end WWII by Christmas 1944 with his Operation Market Garden. Drop airbourn divisions behind the German lines in Holland to capture and hold bridges for the British ground forces to advance rapidly from west of Endhoven to Arnhem and thus outflank the Siegfried Line and overrun the Ruhr valley. Both American and British troops are involved. Can be played in full in about 4 hours (single player) or in 5 different parts requiring less time. Very detailed with each day having 3 phases. It can be won. Hint: Keep everything moving east and once you capture the Arnhem bridge dig in and defend and use that artillery.

In the Works: Bonaparte--the Napoleon capaigns from Italy to Waterloo for 1 to 6 players.